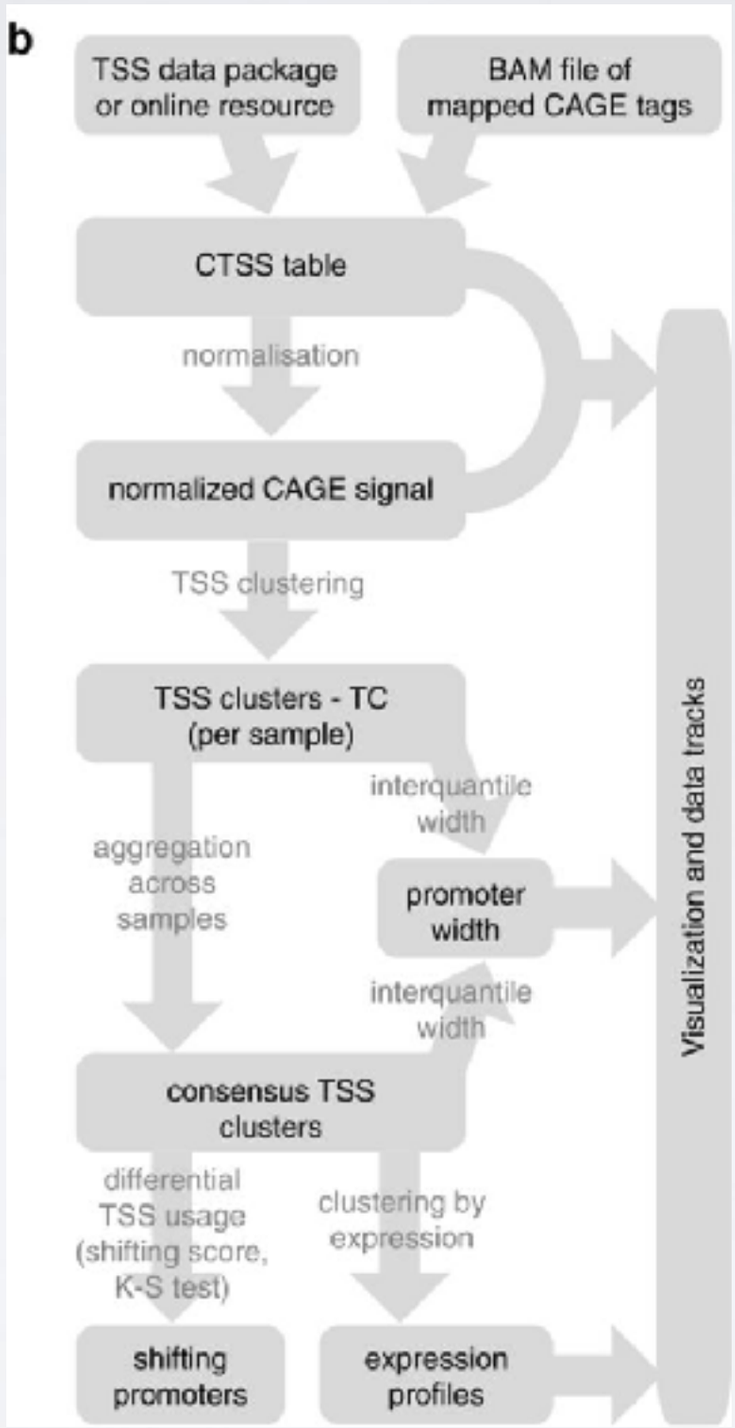
CAGER R PACKAGE

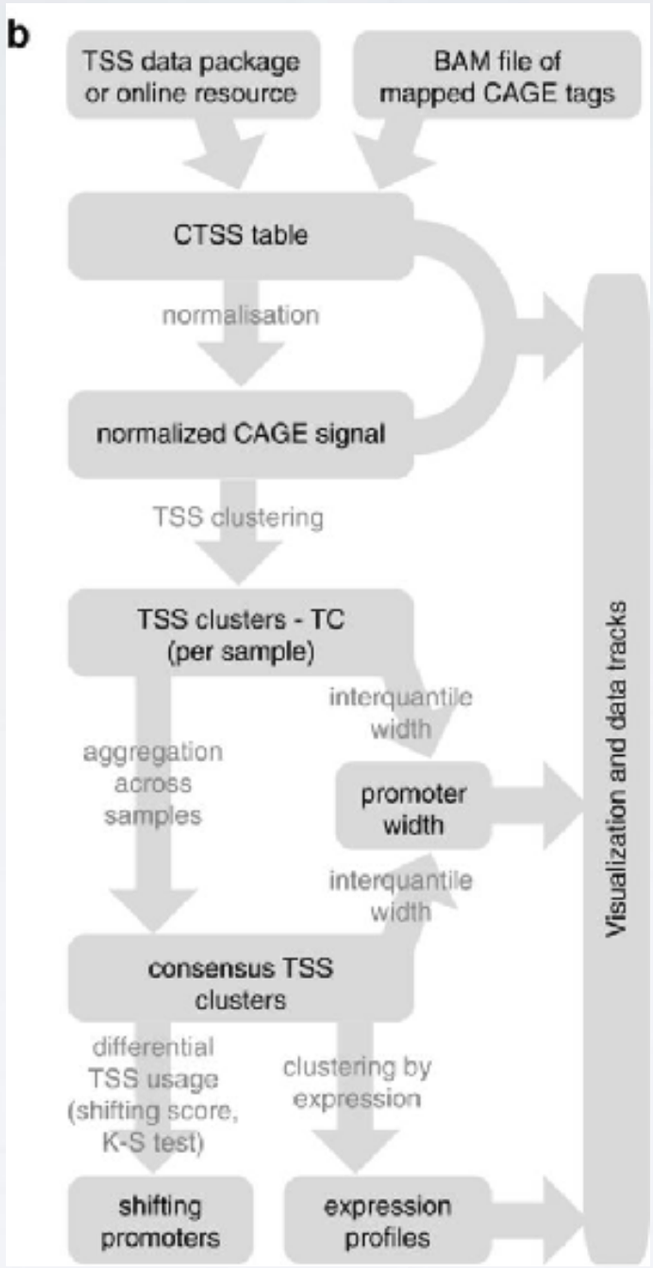
Exploring all its (current) functions

30th August 2017
Leonie Roos
l.roos@lms.mrc.ac.uk

OVERVIEW CAGER WORKFLOW



CAGER WORKFLOW



Haberle V, _et al_. CAGEr: precise TSS data retrieval and high-resolution promoterome mining for integrative analyses. Nucl Acids Res 2015;43(8):e51.

STARTING POINT

I prepared some code

this is found at: https://github.com/leonieroos/CAGEr-F6-workshop

Download the whole dir and save

- open the directory and go to tutorial and open the .rmd file into R studio

DATA FORMATS

CAGEr accepts multiple formats

I CAGE tags mapped to genome

- BAM & BED files

II CTSS files

- Tab separated files with genomic coordinates and number of tags for each CTSS

III CAGE datasets from R packages

- FANTOM5/4/3

From all these, we can create a **CAGEset object**

This is the basis from which the CAGEr functions all work

DATA FORMATS

CAGEr accepts multiple formats

I CAGE tags mapped to genome

- BAM & BED files

II CTSS files

- Tab separated files with genomic coordinates and number of tags for each CTSS

III CAGE datasets from R packages

- FANTOM5/4/3

From all these, we can create a **CAGEset object**

This is the basis from which the CAGEr functions all work

CREATING A CAGESET

Let's start!

Fantom6 Data is in BED Format.. but not *one row per tag*

Easily solved by selecting the columns of:

chromosome, end, strand, and column number 5 (amount of tags per position)

CTSS file

chr1	101	+	5
chr1	104	+	2

line 53

CREATING A CAGESET

Let's start!

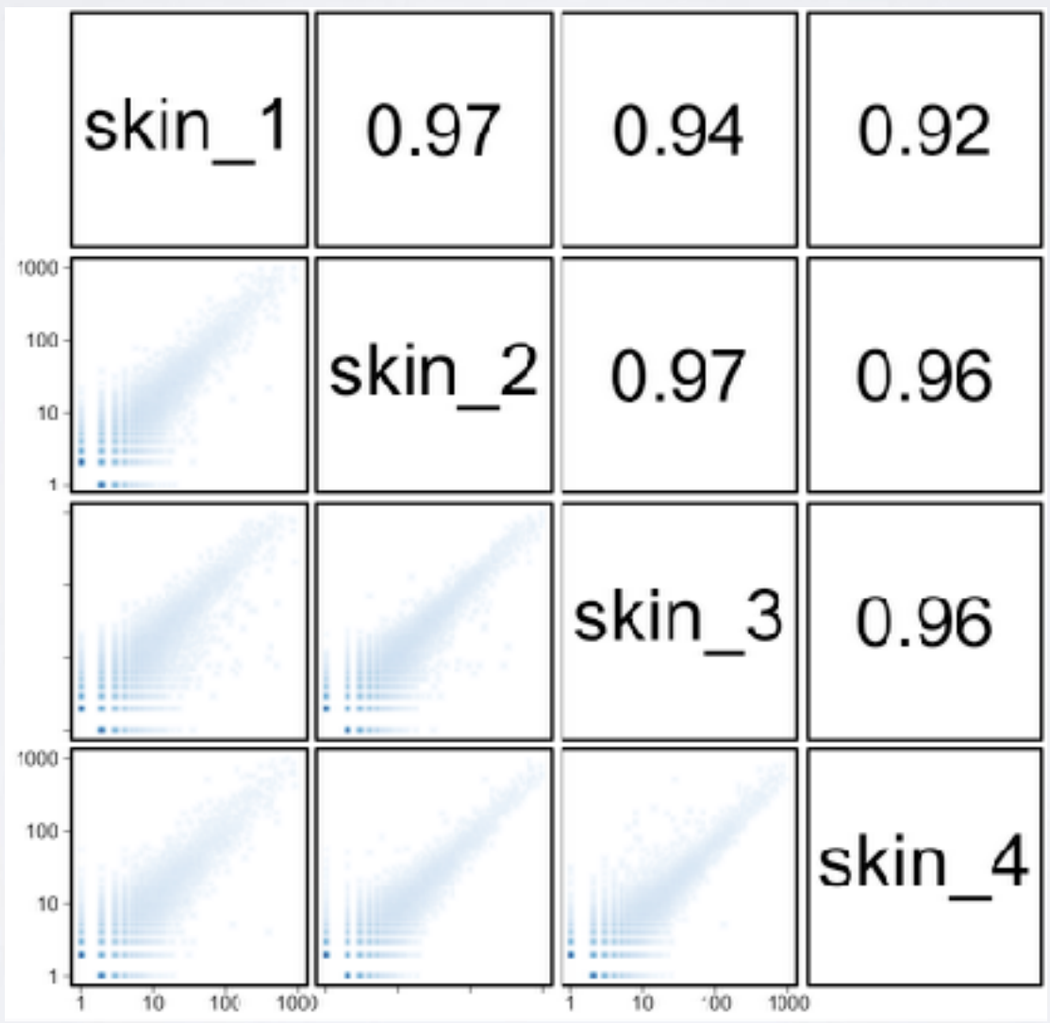
```
### load the CAGEr package
library(CAGEr)
### BSgenome with the right version
library(BSgenome.Hsapiens.UCSC.hg38)

### define where the ctss.bed files provided are located for CAGEr
# where the files can be found
pathsToInputFiles <- list.files("../data/ctss_tables", full.names = TRUE)

### creating a CAGEset object
myCAGEset <- new("CAGEset", genomeName = "BSgenome.Hsapiens.UCSC.hg19", inputFiles = path
sToInputFiles, inputFilesType = "ctss", sampleLabels = paste("skin_",1:4, sep = ""))

# you can check the object:
myCAGEset
```

```
### reading in the data
getCTSS(myCAGEset)
# get a dataframe of ctss counts:
ctss <- CTSStagCount(myCAGEset)
head(ctss)
```

CORRELATION BETWEEN SAMPLES

line 97

```
### creating a correlation plot and table of the samples
corr.m <- plotCorrelation(myCAGEset, samples = "all", method = "pearson")
```



NORMALISATION

Library size differ between samples

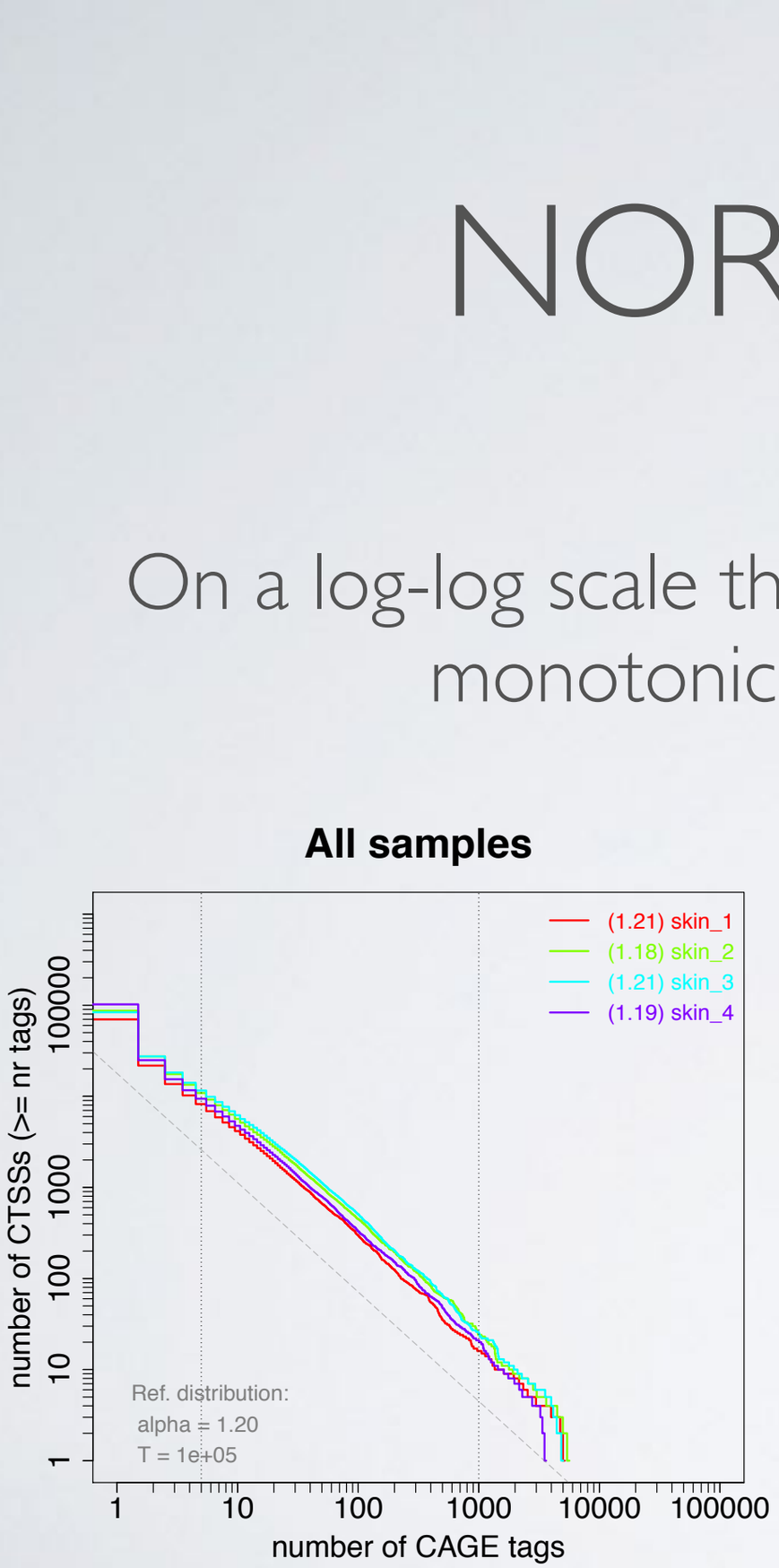
To make samples comparable, we will need to normalise our data.

- Tags per million normalization
- power-law based normalization

Many CAGE-seq data follow a power-law distribution.

NORMALISATION

On a log-log scale this reverse cumulative distribution has a monotonically decreasing linear function



$y = -1 * alpha * x + beta$

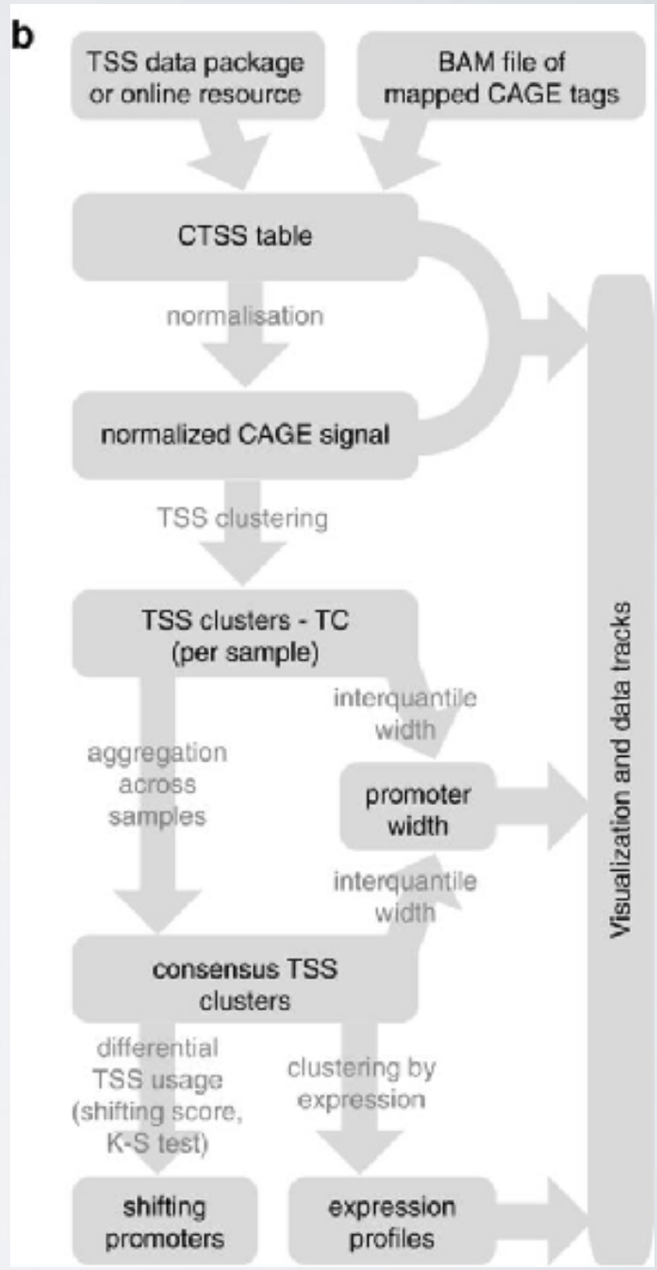
what we need:

- the slope
- the number of tags

Balwierz PJ, Carninci P, Daub CO, et al. Methods for analyzing deep sequencing expression data: constructing the human and mouse promoterome with deepCAGE data. Genome Biology. 2009;10(7):R79.

UP TO NOW

- Imported F6 data into a CAGEset object
- Correlation of CTSS per samples
- Normalised data



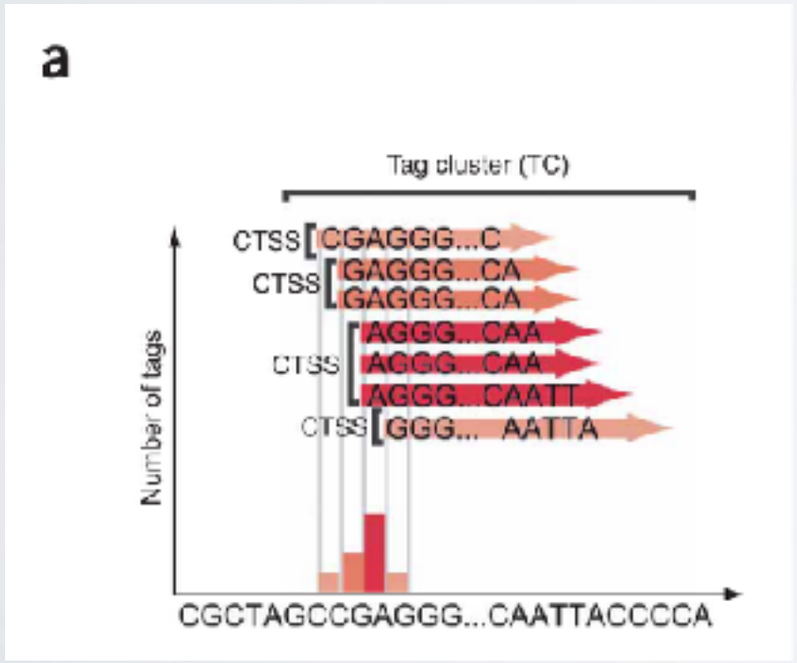
TAG CLUSTERS

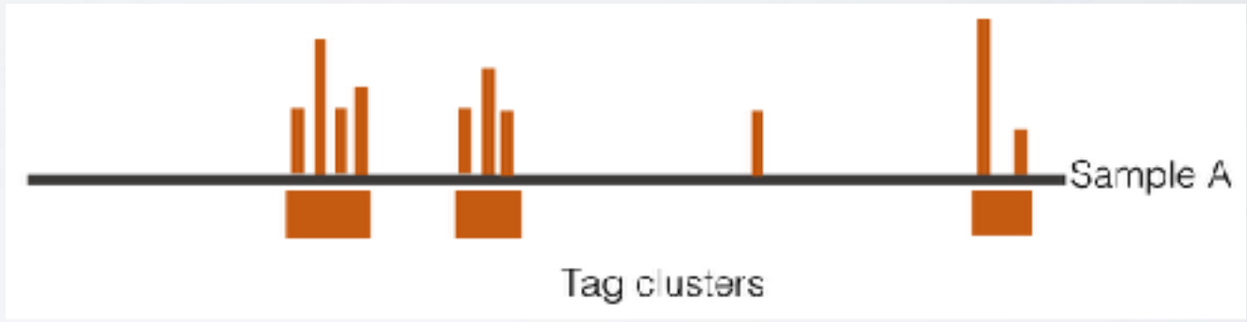
CAGE TSS (CTSS)

- CAGE tags with an identical CAGE-tag starting site.

Tag cluster (TC)

- CTSSs in close proximity (same strand)





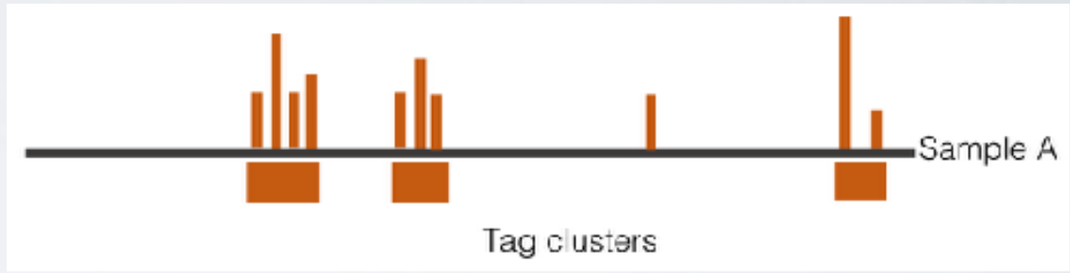
line 126

Carninci P, _et al_. Genome-wide analysis of mammalian promoter architecture and evolution. Nat Genet 2006;38, 626-635.

TAG CLUSTERS

Tag clusters (TC)

- Max distance between CTSSs is 20 bp
- Single CTSS are allowed if > 5 normalised signal




```
clusterCTSS(object = myCAGEset,
            threshold = 1,
            thresholdIsTpm = TRUE,
            nrPassThreshold = 1,
            method = "distclu",
            maxDist = 20,
            removeSingletons = TRUE,
            keepSingletonsAbove = 5)
```

TAG CLUSTER-WIDTH

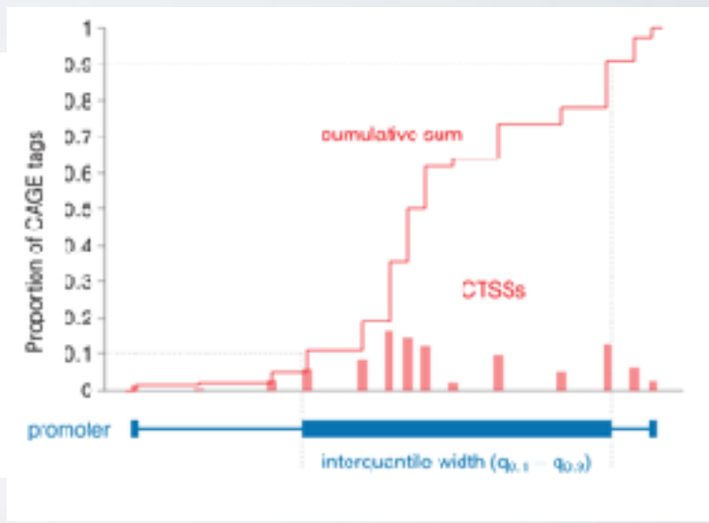
The width of each tag cluster per sample.

To have a width more robust to low tag count outliers:

- Width based on a set of quantiles of the cumulative distribution of tag signal per TC.

Generally, the width of a TC is set between the quantiles 0.1 and 0.9 to capture 80% of CTSS within the TC.

```
cumulativeCTSSdistribution(myCAGEset, clusters = "tagClusters")
quantilePositions(myCAGEset, clusters = "tagClusters", qLow = 0.1, qUp = 0.9)
plotInterquantileWidth(myCAGEset,
                       clusters = "tagClusters",
                       tpmThreshold = 3,
                       qLow = 0.1,
                       qUp = 0.9)
```

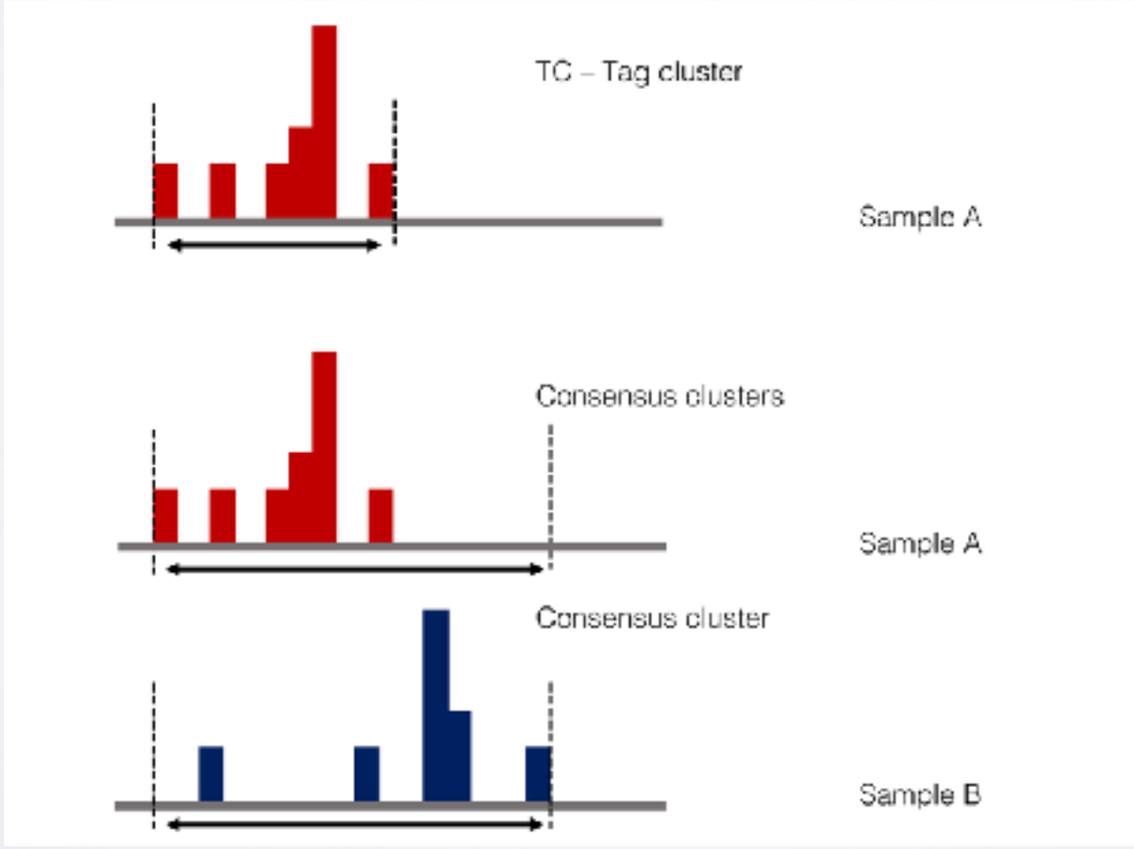


These are the most time consuming steps!

CONSENSUS CLUSTERS

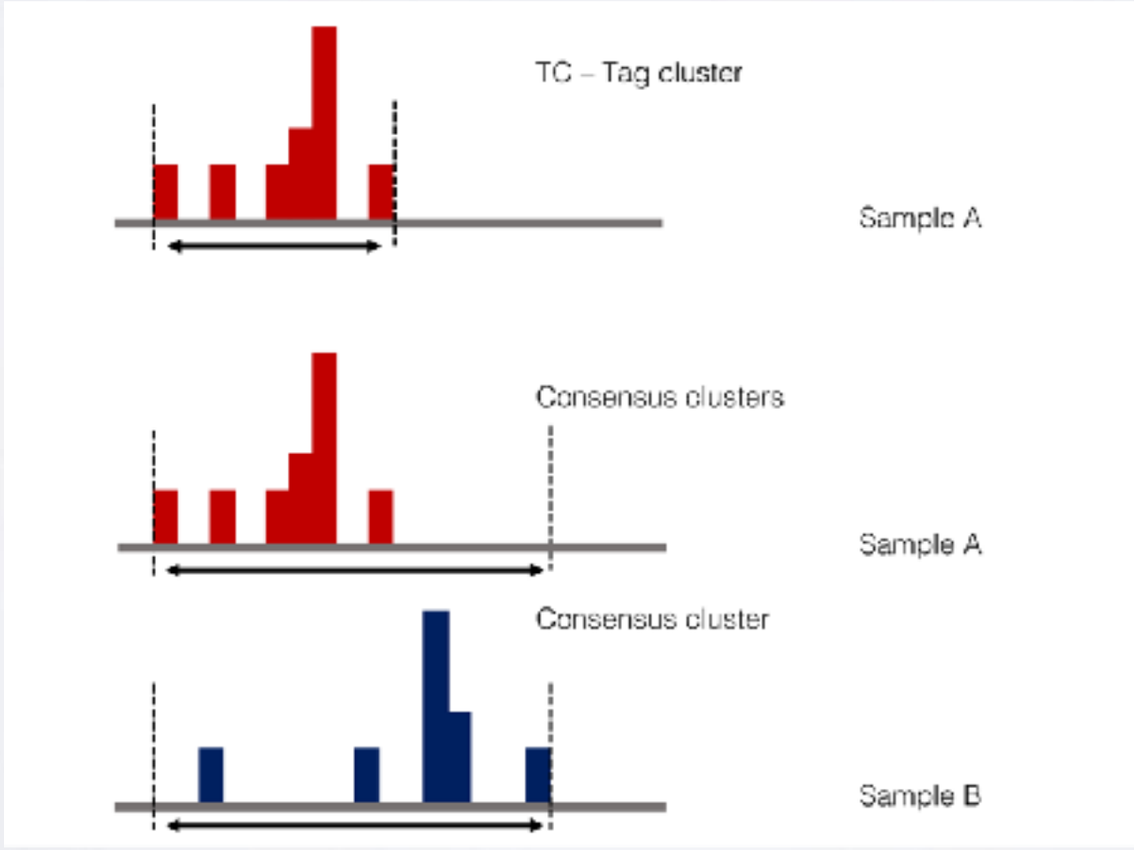
- In many cases TCs do not coincide perfectly
- Maybe two TCs in one sample and one large in other

what if you want to compare TC clusters across samples?

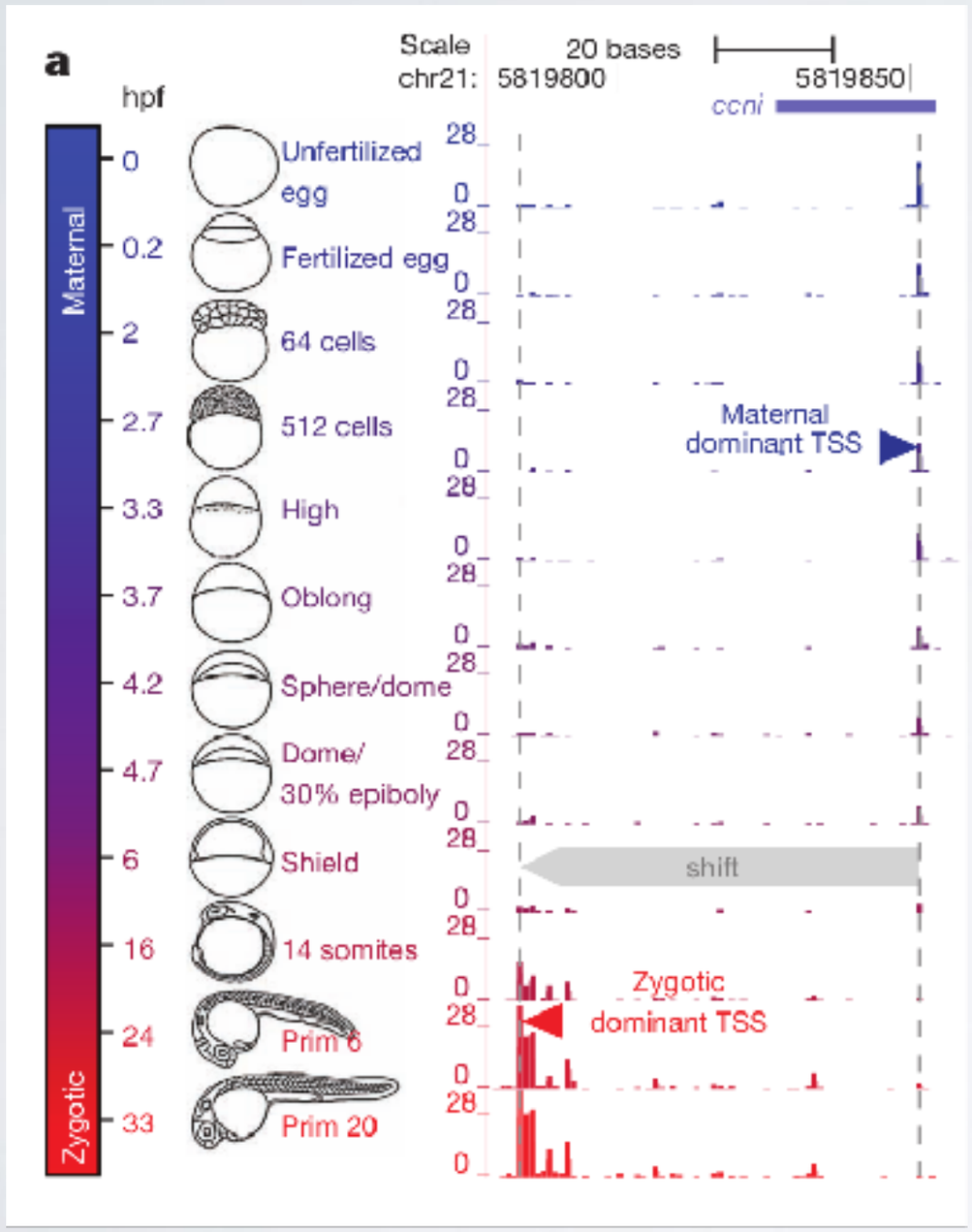


CONSENSUS CLUSTERS

```
aggregateTagClusters(myCAGEset,
                     tpmThreshold = 5,
                     qLow = 0.1,
                     qUp = 0.9,
                     maxDist = 100)
```



PROMOTER SHIFTING

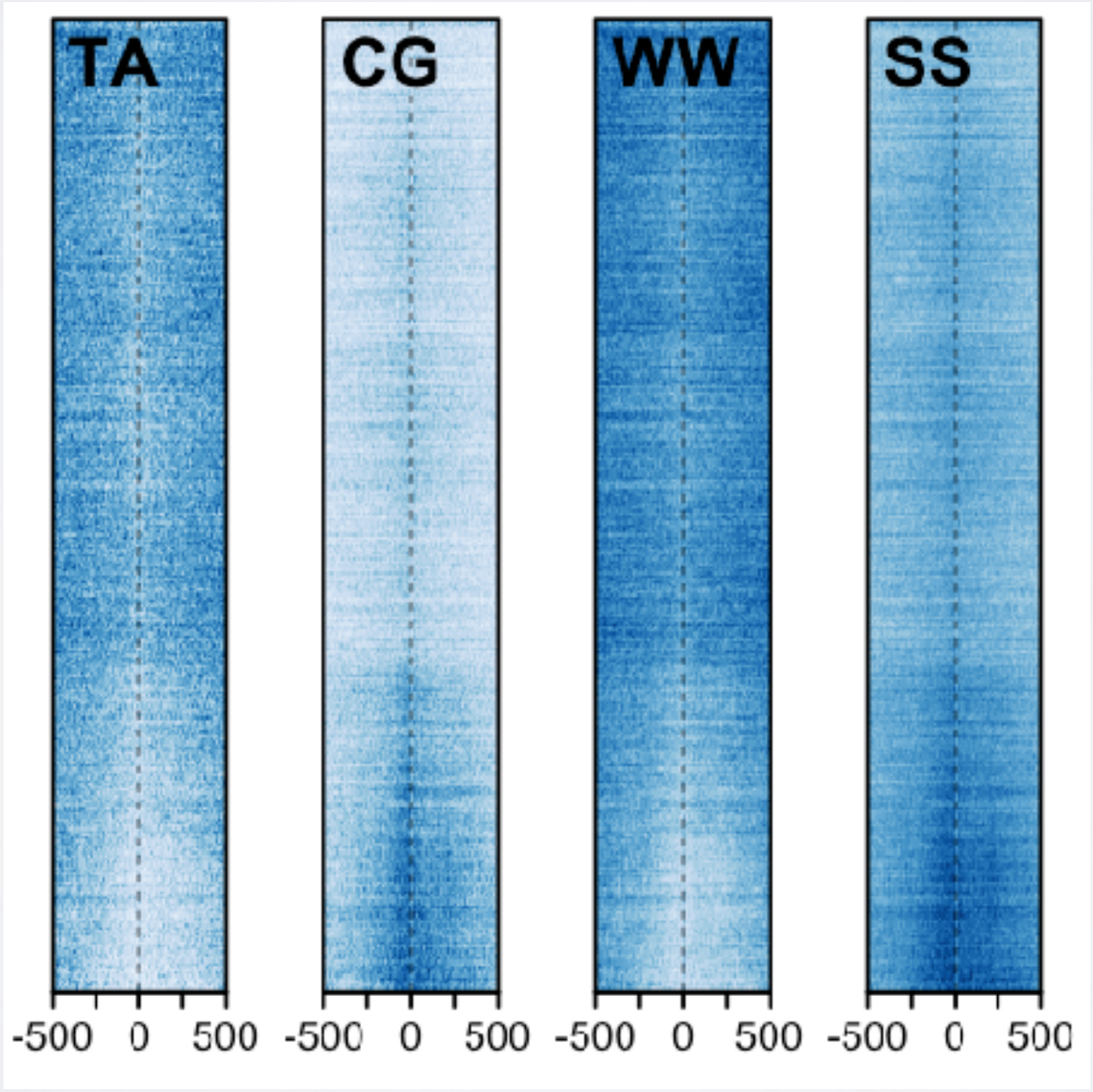


This method has also been implemented in CAGEr

Haberle V, _et al_. Two independent transcription initiation codes overlap on vertebrate core promoters. Nat 2014;507(7492):381-385.

DOMINANT TSS

cluster	chr	start	end	strand	nr_ctss	dominant_ctss	tpm	tpm.dominant_ctss	q_0.1	q_0.9	interquantile_width
1	chr3	3126907	3127018	+	46	3126949	233.2163609	63.32701553	3126937	3126963	27



PROMOTER SHIFTING

The method from haberle *et al* (2014) has also been implemented in CAGEr

- Shifting is detected using cumulative distribution per sample CAGE signal
- A shifting score is determined by the difference in cumulative distributions
- A Kolmogorov-Smirnov is performed to give a general assessment of differential TSS usage